1. (a) (c) (d)
2. (a) (c) (d)
3. (b)
4. (c)
5. (b)
6. (b)
7. (a)
8. (c)
9. (d)
10. (a)
11. **DENORMALIZATION**
    - Denormalization is a database optimization technique in which we add redundant data to one or more tables. This can help us avoid costly joins in a relational database. Note that *denormalization* does not mean ‘reversing normalization’ or ‘not to normalize’. It is an optimization technique that is applied after normalization.
    - Basically, the process of taking a normalized schema and making it non-normalized is called denormalization, and designers use it to tune the performance of systems to support time-critical operations.
    - In a traditional normalized database, we store data in separate logical tables and attempt to minimize redundant data. We may strive to have only one copy of each piece of data in a database.
    - For example, in a normalized database, we might have a Courses table and a Teachers table. Each entry in Courses would store the teacher ID for a Course but not the teacher Name. When we need to retrieve a list of all Courses with the Teacher’s name, we would do a join between these two tables.
12. **DATABASE CURSOR**
    - A database cursor is an identifier associated with a group of rows. It is, in a sense, a pointer to the current row in a buffer.
    - You must use a cursor in the following cases:
        - Statements that return more than one row of data from the database server:
            - A SELECT statement requires a select cursor.
            - An EXECUTE FUNCTION statement requires a function cursor.
    - An INSERT statement that sends more than one row of data to the database server requires an insert cursor.
13. **TYPES OF SQL QUERIES:**

    The following are the most commonly used SQL Queries
    - Creating a table (CREATE TABLE)
    - Inserting records in a table (INSERT INTO)
    - Viewing all records from a table (SELECT)
    - Arranging the records in a table (ORDER BY)
    - Viewing only selected records from a table (SELECT COUNT)
    - Deleting records from a table (DELETE)
14. **CONSTRAINTS**

    SQL constraints are used to specify rules for the data in a table. Constraints are used to limit the type of data that can go into a table. This ensures the accuracy and reliability of the data in the table. If there is any violation between the constraint and the data action, the action is aborted.

    The following are the types of constraints
    - NOT NULL constraints
    - Unique constraints
    - Primary key constraints
    - (Table) Check constraints

- Foreign key (referential) constraints
- Informational constraints.

15. **AUTO INCREMENT**

    Auto-increment allows a unique number to be generated automatically when a new record is inserted into a table. Often this is the primary key field that we would like to be created automatically every time a new record is inserted.